



# Adatbányászat a ClusterGriden

*Vida Gábor*  
MTA SZTAKI

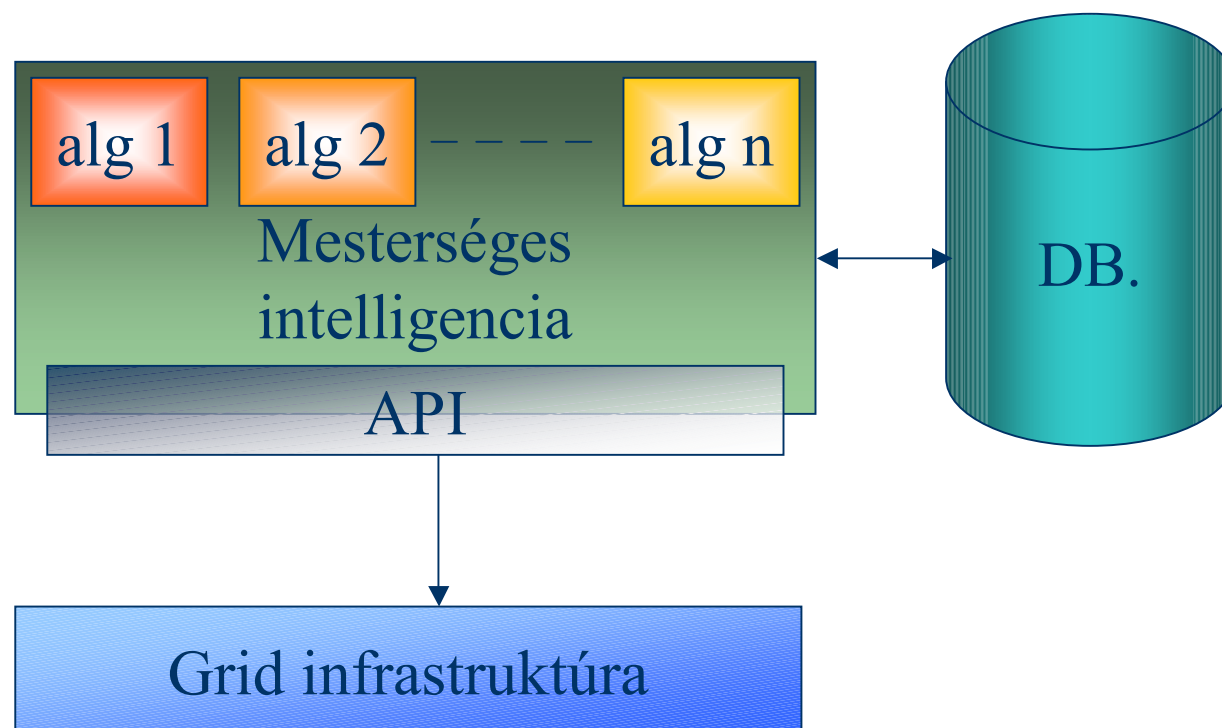
---



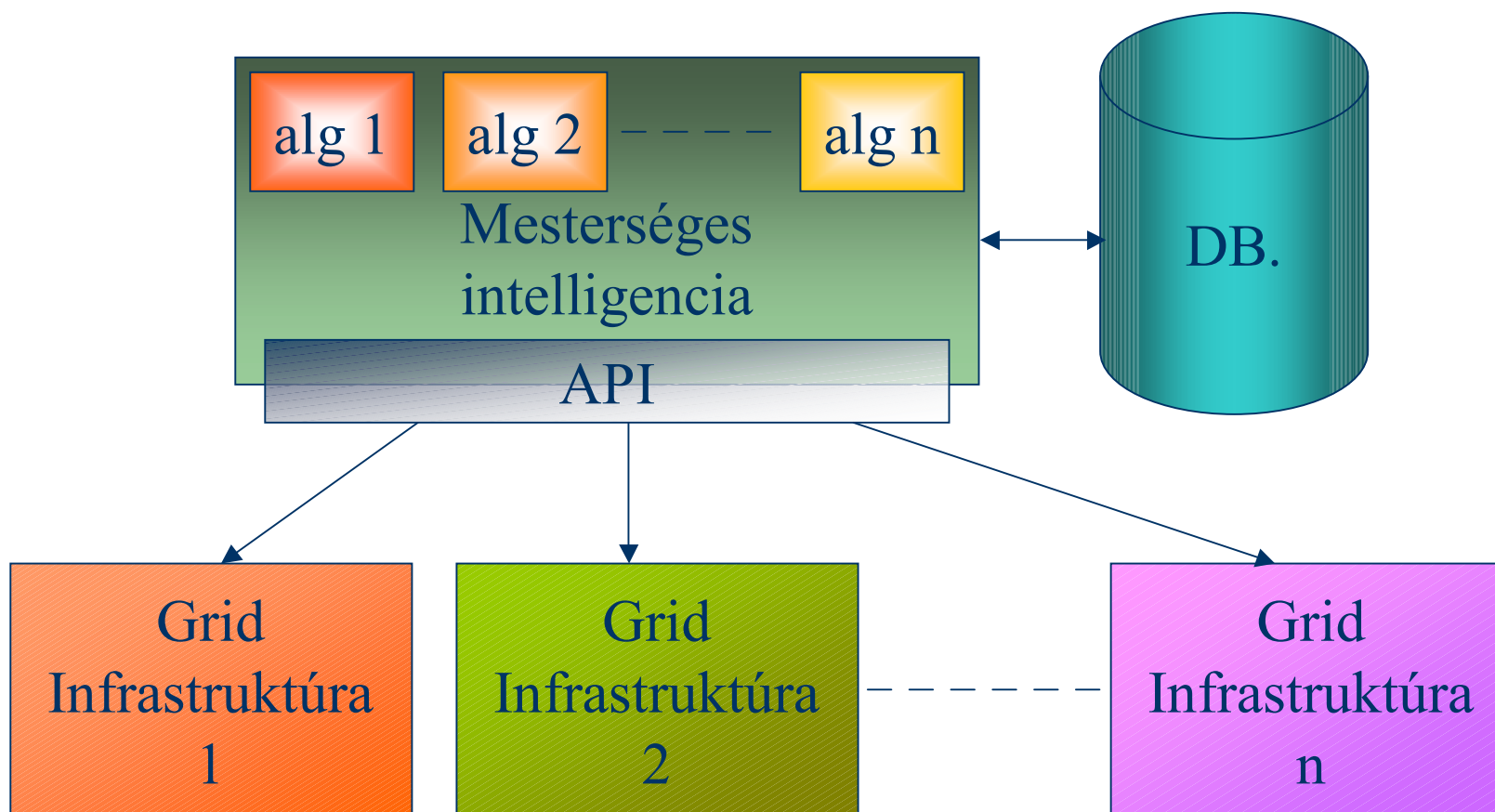
## A történet kezdete

- “Következő generációs adatbányászat nagy teljesítményű elosztott párhuzamos rendszereken”
- *Partnerek:*
  - MTA SZTAKI
  - Szegedi Tudományegyetem,  
Mesterséges Intelligencia Csoport
  - AAM Informatikai Tanácsadó Rt.
  - T-Systems Magyarország

# Prototípus alkalmazás



# Hordozhatóság



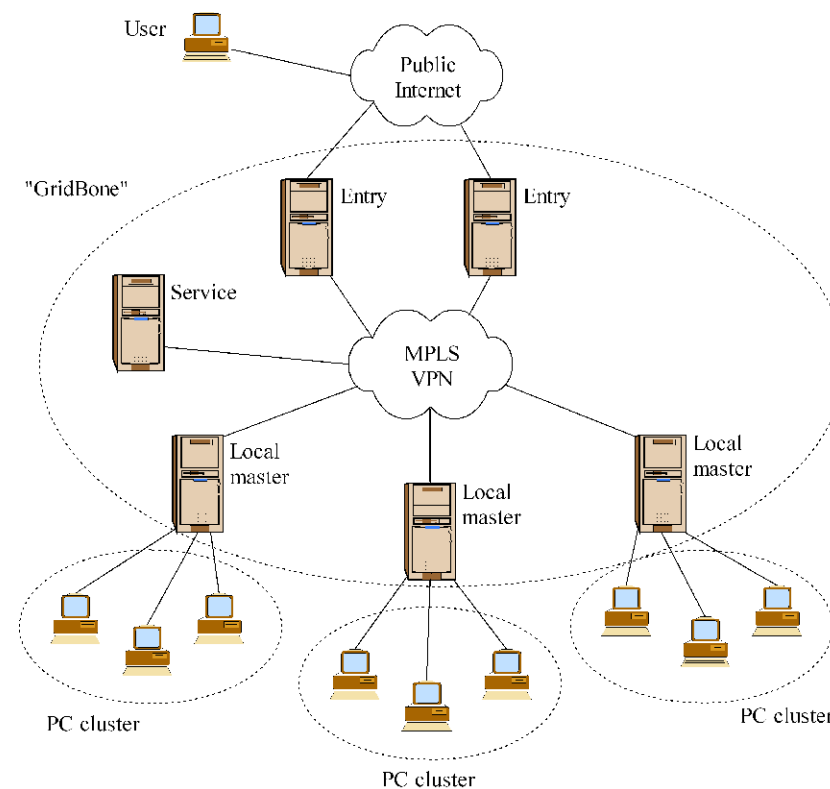


# Infrastruktúra választás

- Az algoritmusok jellemzői alg 1 alg 2 – – alg n
  - Hosszú futási idejűek
  - Futás-közbeni részeredményeik használhatóak
- A Grid rendszerrel szemben támasztott követelmények:
  - Nagy számítási kapacitás
  - Részeredmények kinyerése a rendszerből
- Választott infrastruktúra: ClusterGrid

# A ClusterGrid felépítése

- NIIFI kezdeményezésére
- felsőoktatási intézmények számára kiírt laboratórium-fejlesztési pályázathoz kapcsolódóan
- éjszakánként, ill. a hétvégeken egymáshoz kapcsolva, mint egy nagy Grid rendszert lehessen alkalmazni.





# Felhasználói API biztosítása

- Feladat: egy egyszerű API létrehozása az adatbányász alkalmazás és a Grid infrastruktúra közé.
- Célja:
  - Az adatbányász alkalmazás fejlesztőinek, ne kelljen a Grid keretrendszer sajátosságaival foglalkoznia
  - Az API esetleges hordozhatósága lehetővé teszi, más Grid rendszerek felhasználását a jövőben



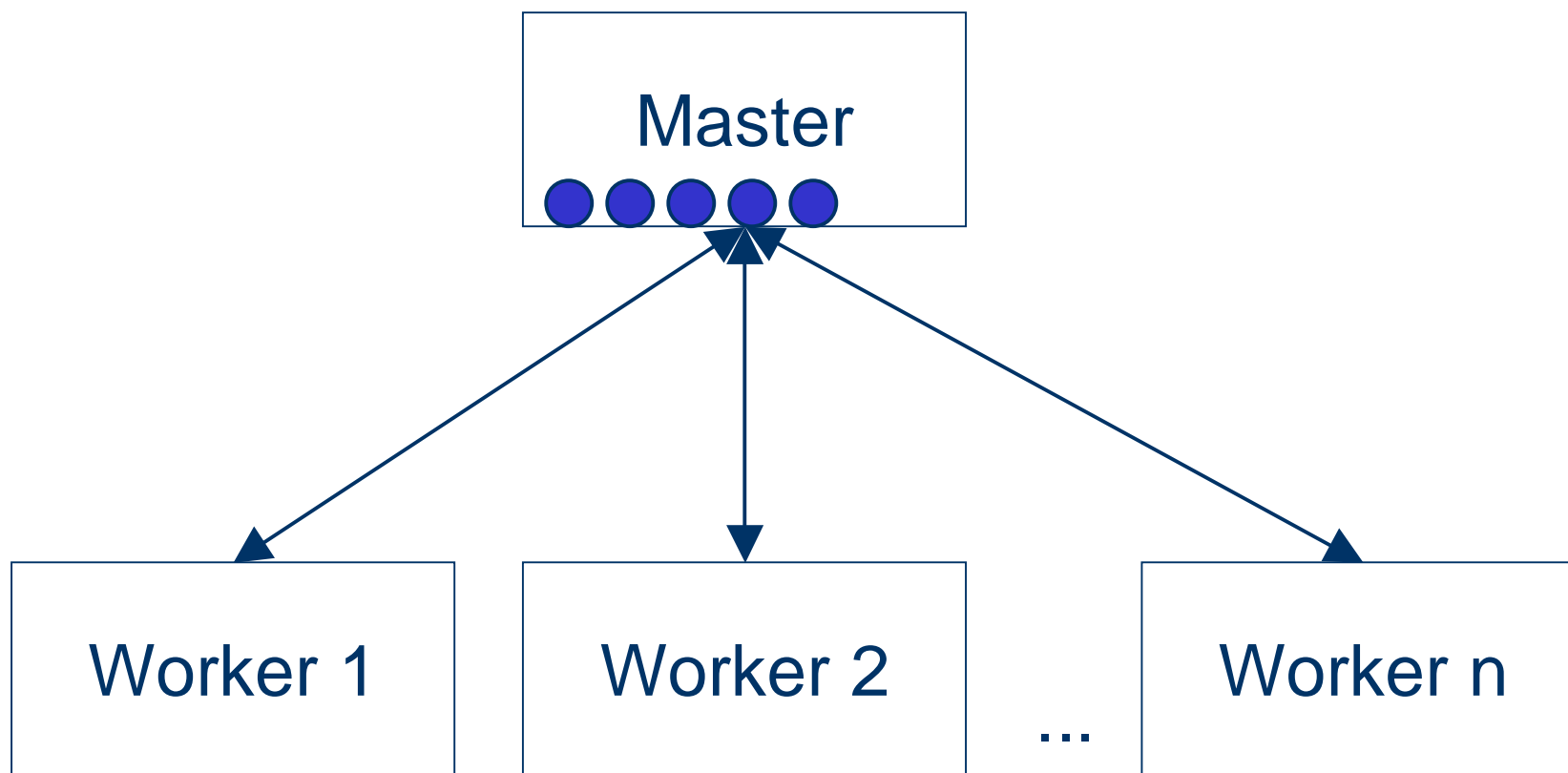
# Az API

- API-val szembeni követelmények:
  - **Egyszerű** felhasználói felület
  - Lehetőség a futó jobok felfüggesztésére
  - Lehetőség a **futás közbeni eredmények** visszajuttatására az alkalmazáshoz
  - Hordozható legyen
- Választás: DC-API (Distributed Computing API)
  - Első használat 2005 nyarán a SZTAKI Desktop Gridben
  - SZTAKI fejlesztés ilyen esetekre
  - Farm jellegű feladatok megoldására



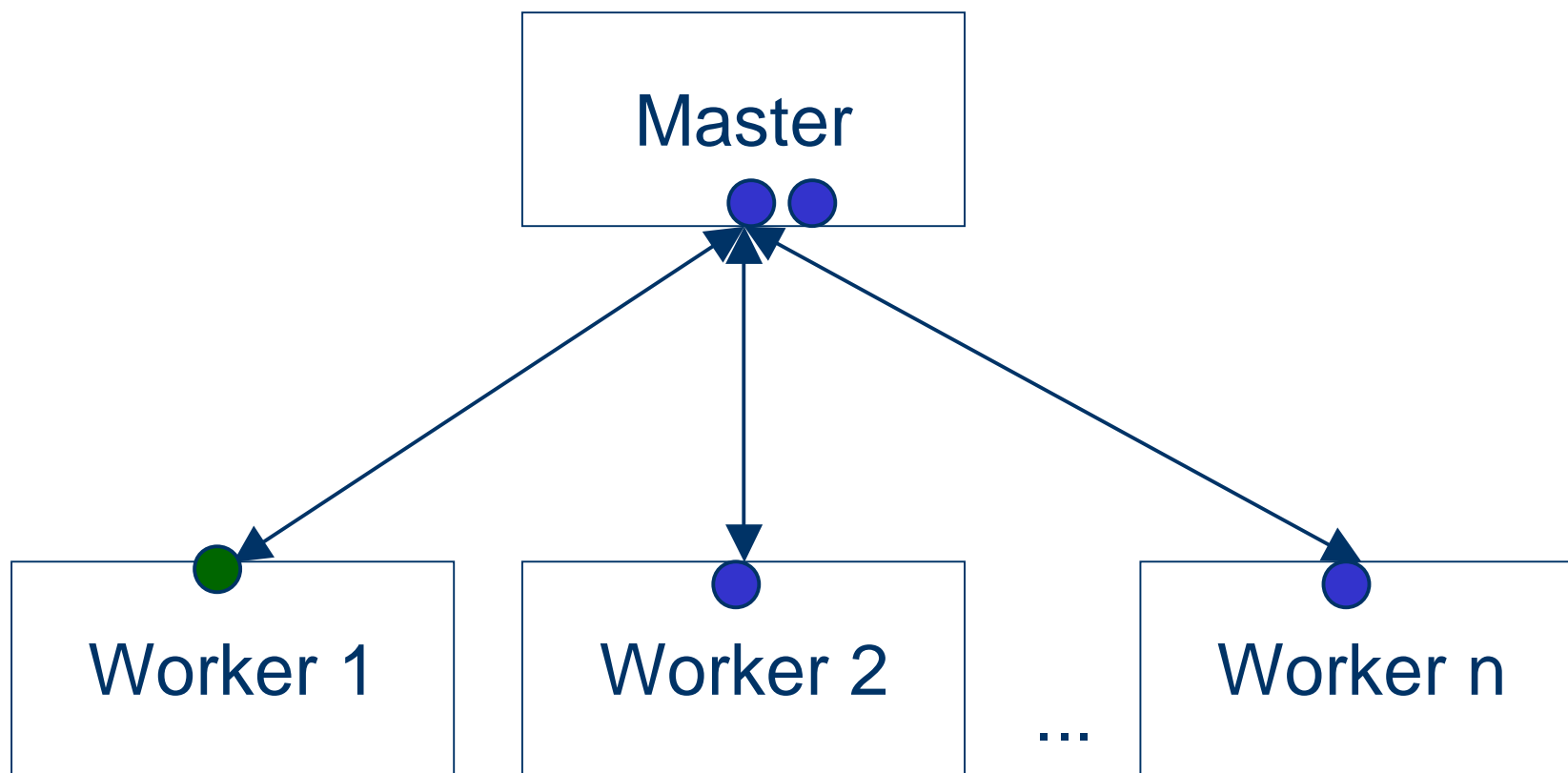


# Farm





# Farm





## Az alkalmazás előkészületei

- A részfeladatokat kiszámító programok önálló végrehajtható állományainak létrehozása.
- Az alkalmazás az egyes részfeladatoknak a bemeneti adatokat egy (vagy több) input fájlba írja ki.
- Eredmény-feldolgozó eljárás fájlból olvassa be az eredményeket.



# DC-API: felépítése

- Részfeladat létrehozása (DC\_createWU)
  - A DC-API egy munkacsomagot (Workunitot azaz WU-t) reprezentáló objektumot hoz létre. Ennek paramétereit különböző DC-API rutinokkal lehet módosítani.
  - Egyedi azonosítóval tér vissza. A későbbiekben ezzel az azonosítóval hivatkozhat az egyes WU-kra a programozó.
- Részfeladat elküldése végrehajtásra (DC\_submitWU)
  - A DC-API létrehozza a ClusterGrid-ben használatos Jobdir könyvtárstruktúrát, majd feltölti a megfelelő értékekkel (argumentum, input fájlok ... stb), és a *clgr\_submit* paranccsal beküldi a jobot végrehajtásra.
- Futó részfeladat megszakítása (DC\_cancelWU)
  - Törli a jobot a ClusterGridből a *clgr\_rm* parancs használatával.
- Eredményre várakozás (DC\_checkForResult)
  - Meghatározott időközönként ellenőrzi a beküldött jobok állapotát (*clgr\_status*), és jelzi az alkalmazásnak, ha valamelyik lefutott, visszaadva az eredményt (*clgr\_getout*).
- Részfeladat teljes törlése (DC\_destroyWU)
  - Törli memóriából a WU adatait és a WU részére kialakított Jobdit könyvtárstruktúrát.



## DC-API: Suspend / Resubmit

- Nem valódi felfüggesztésről van szó
  - Csak szimulálja az alkalmazás felé
- A DC-API leállítja a job futását (*clgr\_rm*), de előtte visszakéri a Grid-től annak aktuális eredményeit. (*clgr\_getout*)
- Amennyiben a kliens programot úgy írták meg, hogy időről-időre készítsen mentést az állapotáról, úgy ebben az esetben egy újbóli beküldés esetén az addig elvégzett munka nem vesz el.

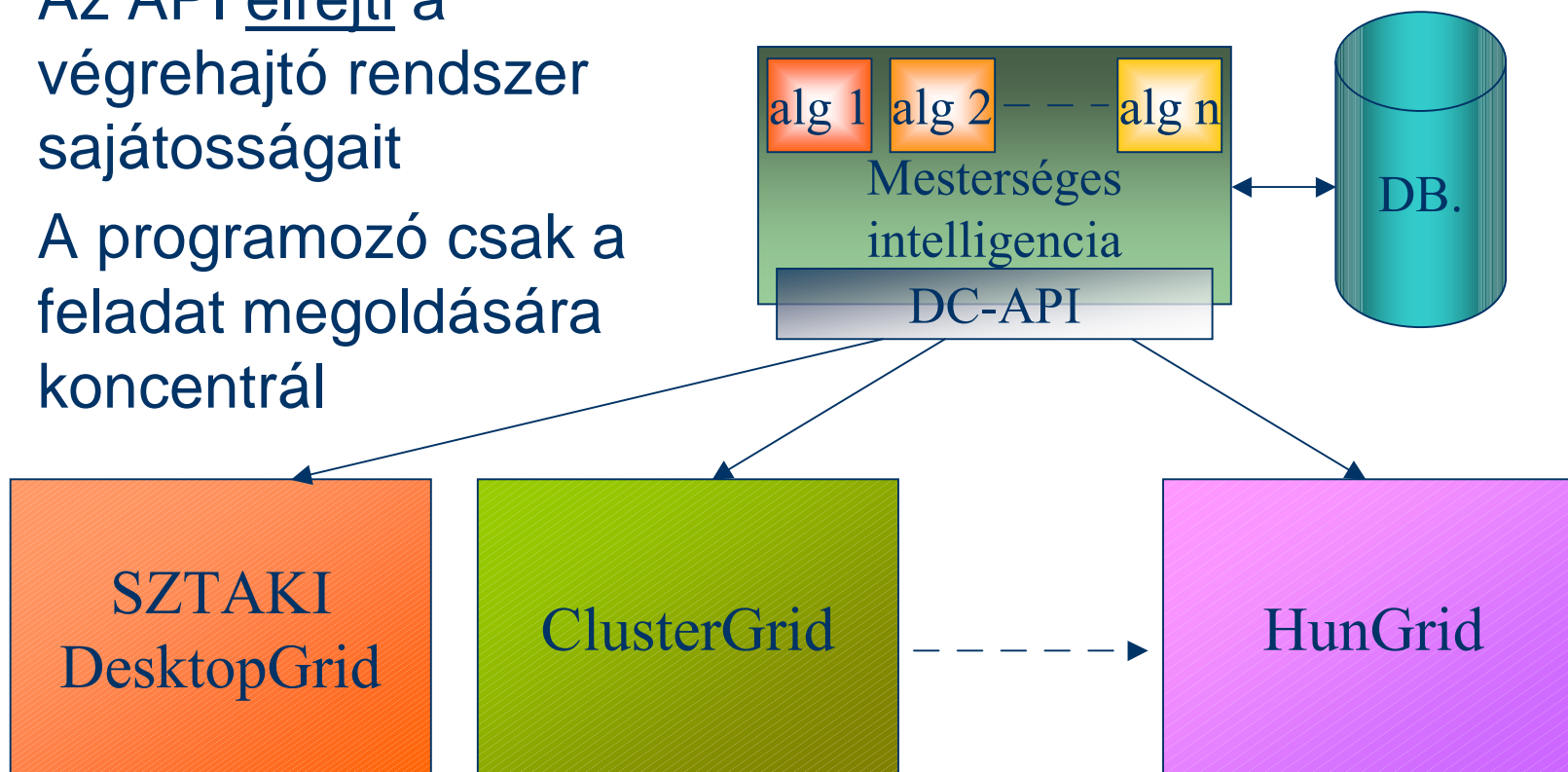


## DC-API: Futás közbeni eredmény

- A worker alkalmazásnak értesítenie kell a DC API-t, hogy használható részeredménnyel rendelkezik a master alkalmazás számára.
- Megoldás: hirdetés készítése (valamilyen plusz információt tudjon hozzájuttatni a státusz listájához), mely jelenleg a SZTAKI-ban kifejlesztett Mercury Grid monitorozó rendszerrel lett megvalósítva.
- A DC-API elegendő, hogy a jobok státusz listáját kérdezze le időről időre, amit úgyis megtenne.

# Hordozhatóság

- Az API elrejt a végrehajtó rendszer sajátosságait
- A programozó csak a feladat megoldására koncentrál





# Összefoglalás

- Adatbányászat a ClusterGridben és máshol
  - A ClusterGridre történő sikeres implementálás csak a kezdet
  - Folyamatban van a Condor-os implementáció és remélhetőleg még sok másik követi a sort ...
- Valamint megkezdődött a DC-API 2. generációs verziójának kifejlesztése
  - Hordozhatóbb
  - Könnyebben használható

**Köszönöm a figyelmet!**